

PENERAPAN PETRI-NET PADA MODEL GERAKAN BERJALAN *TROTTING* ROBOT BERKAKI EMPAT (*QUADRUPED*)

Tony Yulianto

Universitas Islam Madura, toni_yulianto65@gmail.com

Abstract. Nowadays many robots has produced not just on a few scale, but also on large for helping human on doing task everyday corresponding the function. By the way, the function can be good functionate if the components, which support to the robot, can be good functionate, too, like the other one is robot walked way movement. Here, we look on four legs of robot walked movement using trotting, because we can find this nowadays and the trouble is kind than robot walked movement with two legs. On this case, we will use max-plus algebra to solve this problem to get robot walked way model on trotting which is like what we want.

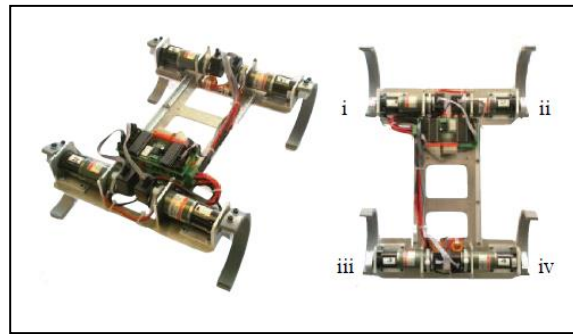
Keywords: *max-plus algebra, trotting way movement, Petri net, four legs robot.*

Abstrak. Selama ini sudah banyak robot diproduksi baik dalam skala kecil maupun besar untuk membantu manusia dalam mengerjakan pekerjaan sehari-hari manusia sesuai dengan fungsi masing-masing. Akan tetapi, fungsi-fungsi tersebut akan dapat berjalan dengan baik apabila komponen-komponen yang mendukung pada robot dapat berjalan dengan baik, seperti salah satunya adalah cara gerak berjalan robot. Di sini, peneliti lebih menekankan gerak jalan robot empat kaki yang menggunakan *trotting*, karena yang banyak umum digunakan dan kesulitannya juga lumayan dibandingkan robot berjalan dengan dua kaki. Dengan menggunakan aljabar max plus akan diperoleh model gerak jalan robot *trotting* tersebut yang sesuai dengan yang diharapkan.

Kata Kunci: *aljabar max plus, gerak jalan trotting, Petri net, robot empat kaki.*

1 Pendahuluan

Pada gerak robot berjalan dari sebuah robot berkaki dua biasa (kontinu), kaki kiri hanya boleh terangkat (*lift off*) dari tanah setelah kaki kanan mendarat (*touched down*), untuk memastikan robot tidak jatuh karena kurangnya penyangga (*support*). Keperluan sinkronisasi ini dapat dimodelkan dengan evolusi dari sistem kejadian diskrit dengan meringkaskan setiap siklus batas pada lingkaran ke dalam dua peristiwa berurutan dalam sirkuit tertutup [5]: lepas landas (*lift off*) dan mendarat (*touchdown*). Tiap kaki kemudian dimodelkan oleh sebuah siklus kejadian dan sinkronisasi antara kaki diberlakukan oleh koneksi antara tiap siklus kaki yang digambarkan dalam Gambar 1. Sinkronisasi dari beberapa kaki adalah sangat penting khususnya untuk robot pendaki, dimana kurangnya penyangga dapat mengakibatkan konsekuensi yang berakibat fatal.

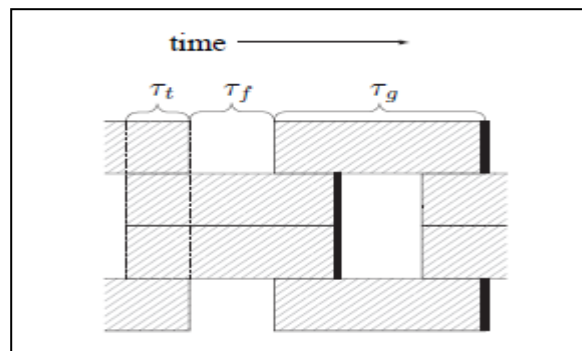


Gambar 1. Sudut pandang dan pandangan dari atas pada robot berkaki empat yang dikembangkan di DCSC, TU Delft, terinspirasi oleh robot *hexapod Rhex*. Nomor romawi pada gambar kanan merepresentasikan urutan penomoran kaki.

2 Hasil dan Pembahasan

2.1 Cara Berjalan *Trotting* Robot Berkaki Empat (*Quadruped Gaits*)

Gambar 2 mengilustrasikan salah satu cara berjalan untuk *quadruped* yakni jalan pelan (*trotting*) [3]. *Trotting*, umumnya digunakan oleh unta dan unta betina, merupakan cara berjalan bilateral, yaitu, kaki kiri depan dan belakang disinkronkan dan berlawanan fase dengan kaki kanan depan dan belakang.



Gambar 2 . Evolusi waktu dari cara berjalan untuk *quadruped*. Kotak yang berarsir merepresentasikan posisi kaki ketika diam di tanah (*stance*), dan garis tebal padat merepresentasikan kejadian *lift off* $l_i(k+1)$. Panah pada a) merepresentasikan kejadian yang harus berlangsung untuk setiap kejadian *lift off* dan *touchdown* agar terjadi

Keterangan dari Gambar 2:

- $t_i(k)$ Waktu *touchdown* untuk kaki ke- i pada iterasi ke- k
- $l_i(k)$ Waktu *lift off* untuk kaki ke- i pada iterasi ke- k
- i Indeks untuk kaki
- τ_f Waktu yang dibutuhkan oleh kaki ketika berada di udara
- τ_g Waktu yang dibutuhkan oleh kaki ketika berada di tanah
- τ_t Parameter pergantian waktu untuk cara berjalan *trotting*

2.2 Graf Kejadian dengan Waktu

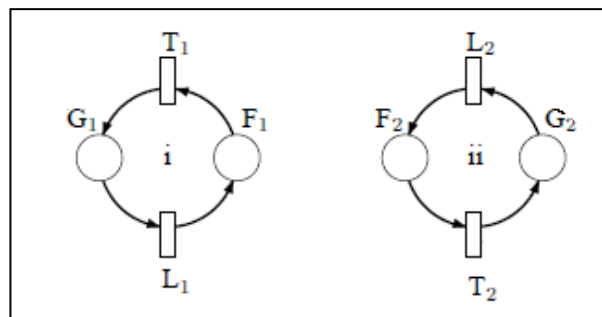
Tulis teori Petri net menyediakan alat intuitif untuk mewakili DES (*Discrete Event System*) secara grafis yang dinamakan graf Petri net [5]. Hal ini mudah untuk menghasilkan cara berjalan dengan mengembangkan suatu Petri net dalam waktu jika: 1) tiap kaki dimodelkan sebagai sebuah sirkuit dengan transisi *lift off* dan *touchdown* 2) sinkronisasi antar kaki diimplementasikan dengan melakukan *fire* kondisi pada transisi *lift off*. Selain itu, dengan merancang Petri net dapat diambil keuntungan dalam hal alat analisis dan sintesis aljabar *max-plus*. Sebuah sub kelas Petri net dengan waktu yang disebut graf kejadian secara resmi setara dengan sistem *linear max-plus* [5]. Hal ini memungkinkan untuk desain intuitif dari cara berjalan oleh graf kejadian dengan terjemahan langsung ke sistem *linear max-plus* dan sebaliknya [4].

Definisi 1. Sebuah Petri net dengan waktu \mathcal{G} dikarakterisasikan oleh sebuah himpunan place \mathcal{P} , sebuah himpunan transisi \mathcal{Q} , sebuah himpunan garis \mathcal{D} dari transisi ke place dan sebaliknya, sebuah tanda awal (initial marking) \mathcal{M}_0 , dan vektor pemegang waktu \mathcal{J} . Jika tiap place memiliki tepat satu transisi upstream dan satu transisi downstream, maka Petri net dengan waktu disebut graf kejadian dengan waktu (*timed event graph*).

Dari Gambar 2 terlihat bahwa kaki *i*, *ii*, *iii*, dan *iv* memiliki waktu *lift off* dan *touchdown* yang berbeda-beda. Oleh karena itu, pada cara berjalan *trotting* ini kaki tidak memiliki waktu kejadian yang sama, sehingga masing-masing kaki bekerja sendiri-sendiri. Asumsikan tiga kaki berada di tanah bersama-sama.

Dengan konstruksi graf kejadian dari cara berjalan *trotting* dihasilkan dari waktu dengan cara berjalan *pacing* adalah sebagai berikut [2]:

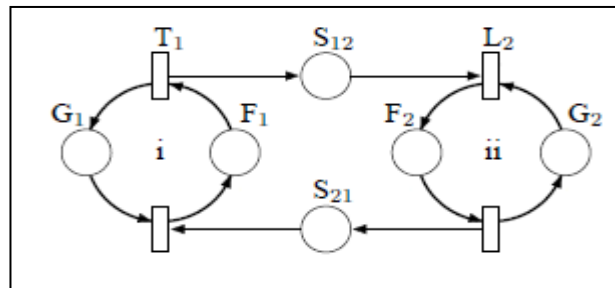
- 1) Untuk tiap kaki, definisikan sebuah sirkuit dengan dua peristiwa (*event*): *touchdown* T_i dan *lift off* L_i . Antar peristiwa, tambahkan place G_i untuk waktu ketika kaki *i* berada di tanah dan F_i untuk saat kaki berada di udara. Hal ini ditunjukkan oleh dua sirkuit tebal bernomor seperti diilustrasikan pada Gambar 3:



Gambar 3. Place G_i untuk waktu ketika kaki *i* berada di tanah dan F_i untuk saat kaki berada di udara.

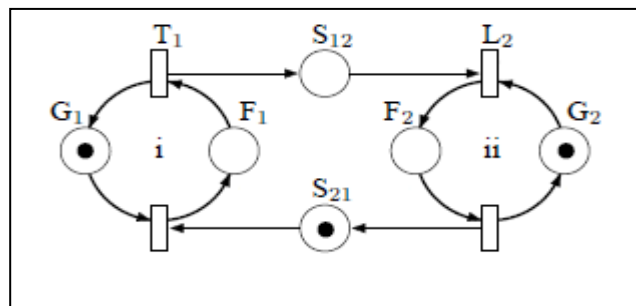
- 2) Untuk tiap sinkronisasi yang dibutuhkan, tambahkan sebuah *place* antara transisi. Hal ini penting untuk memastikan bahwa tiap *place* memiliki sebuah garis *upstream* yang unik dan sebuah garis *downstream* yang unik. Sebagai

contoh, *place* berlabel S_{12} menentukan bahwa transisi *lift off* kaki 'ii' hanya boleh di-fire setelah kaki 'i' telah menyentuh tanah. Hal ini diilustrasikan oleh Gambar 4:



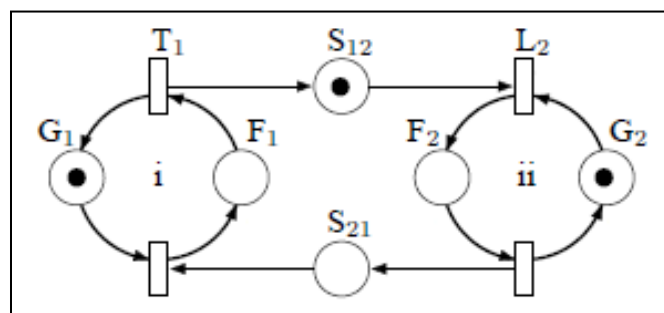
Gambar 4. Place antara transisi.

- 3) Inisialisasi tanda sedemikian hingga semua *place* yang menyatakan kaki sedang berada di tanah (*ground place*) memiliki sebuah token (robot memulai gerakan dengan semua kaki berada di tanah). Selain itu, token ditambahkan pada *place* tersinkronisasi sedemikian rupa sehingga semua sirkuit (tertutup) dapat hidup dan tanda awal dapat dilaksanakan. Hal ini dapat diilustrasikan oleh Gambar 5 dan 6:



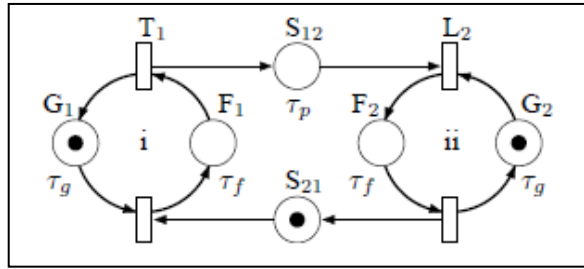
Gambar 5. Inisialisasi tanda jika S_{21} dilakukan.

jika kaki yang akan terangkat terlebih dahulu adalah kaki 'i' dan



Gambar 6. Inisialisasi tanda jika S_{12} dilakukan.

jika kaki yang akan terangkat terlebih dahulu adalah kaki 'ii'.
Jadi secara lengkapnya, graf kejadian dengan waktu atau Petri net dengan waktu untuk cara berjalan *trotting* yang bersumber dari cara berjalan *pacing* dapat digambarkan pada Gambar 7:



Gambar 7. Graf kejadian dengan waktu untuk cara berjalan *trotting*.

2.3 Model Aljabar *Max-Plus* Pada Petri-net Cara Berjalan *Trotting*

Mengikuti langkah-langkah yang disajikan di atas, adalah mungkin untuk mereproduksi banyak cara berjalan. Setelah graf kejadian dengan waktu dibangun, akan sangat mudah untuk menemukan sistem *linear max-plus* yang terkait [4]: κ_j Ψ_i ψ_i v_j

- 1) Tiap transisi Ψ_i dalam graf kejadian dengan waktu diberikan ke sebuah variabel kejadian ψ_i dalam aljabar *max-plus*.
- 2) Untuk tiap garis masuk ke transisi Ψ_i buat daftar dari
 - Kejadian Ψ_j yang mendahului kejadian Ψ_i untuk garis tersebut dan tambahkan j ke dalam suatu himpunan \mathcal{S}
 - Konstanta waktu v_j dari *place* asal garis ini
 - jumlah token κ_j dari *place* asal garis ini
 Kemudian tulis ekspresi: T_1

$$\psi_i(k + 1) = \max_{j \in \mathcal{S}} (\psi_j(k + 1 - \kappa_j) + v_j) \tag{1}$$

Sebagai contoh, lihat transisi T_2 pada Gambar 3. Transisi ini memiliki garis masuk tunggal dari transisi L_2 dengan konstanta waktu τ_f dan nol token. Dengan demikian, Persamaan 1 untuk transisi ini adalah

$$t_2(k + 1) = l_2(k + 1) + \tau_f$$

Untuk transisi L_2 terdapat dua garis masuk dari T_2 dan T_3 , dengan masing-masing konstanta waktu (τ_g, τ_t) dan satu token per *place*. Dengan demikian, persamaan transisi untuk l_1 adalah:

$$l_2(k + 1) = \max(t_1(k) + \tau_t, t_2(k) + \tau_g, t_4(k) + \tau_t)$$

Mengikuti prosedur ini, maka diperoleh persamaan transisi untuk cara berjalan *trotting* yakni:

$$\begin{aligned} t_2(k + 1) &= l_2(k + 1) + \tau_f \\ l_2(k + 1) &= \max(t_1(k) + \tau_t, t_2(k) + \tau_g, t_4(k) + \tau_t) \\ t_3(k + 1) &= l_3(k + 1) + \tau_f \\ l_3(k + 1) &= \max(t_1(k) + \tau_t, t_3(k) + \tau_g, t_4(k) + \tau_t) \end{aligned} \tag{2}$$

Himpunan Persamaan 2 dapat diubah ke dalam struktur persamaan $x(k+1) = A_{(m(k))} \otimes x(k)$, dengan substitusi rekursif [3]. Sebagai contoh,

Persamaan 2 diupdate untuk t_1 menjadi:

$$\begin{aligned} t_2(k+1) &= l_2(k+1) + \tau_f \\ &= \max(t_1(k) + \tau_t, t_2(k) + \tau_g, t_4(k) + \tau_t) + \tau_f \\ &= \max(t_1(k) + \tau_f + \tau_t, t_2(k) + \tau_f + \tau_g, t_4(k) + \tau_f + \tau_t) \end{aligned} \quad (3)$$

Mengikuti proses rekursif yang sama pada Persamaan 2 dan 3 untuk variabel lainnya pada cara berjalan *trotting*, dan mendefinisikan variabel keadaan $x(k) \in \mathbb{R}_{max}^4$ dengan

$$x(k) = [t_1(k) \quad l_1(k) \quad t_2(k) \quad l_2(k) \quad t_3(k) \quad l_3(k) \quad t_4(k) \quad l_4(k)]^T,$$

dapat diperoleh matriks sistem *linear-max-plus* untuk cara berjalan *trotting*, yang dinotasikan A_t . Sehingga persamaan transisi untuk berjalan *trotting* diperoleh [1]:

$$\begin{aligned} t_1(k+1) &= l_1(k+1) + \tau_f \\ l_1(k+1) &= \max(t_1(k) + \tau_g, t_2(k+1) + \tau_g, t_3(k+1) + \tau_g, t_4(k) + \tau_g) \\ \\ t_2(k+1) &= l_2(k+1) + \tau_f \\ l_2(k+1) &= \max(t_1(k) + \tau_t, t_2(k) + \tau_g, t_4(k) + \tau_t) \\ \\ t_3(k+1) &= l_3(k+1) + \tau_f \\ l_3(k+1) &= \max(t_1(k) + \tau_t, t_3(k) + \tau_g, t_4(k) + \tau_t) \\ \\ t_4(k+1) &= l_4(k+1) + \tau_f \\ l_4(k+1) &= \max(t_1(k) + \tau_g, t_2(k+1) + \tau_g, t_3(k+1) + \tau_g, t_4(k) + \tau_g) \end{aligned} \quad (4)$$

Untuk cara berjalan simetri diasumsikan bahwa

$$\tau_g > \tau_f, \text{ dan } \tau_t = \frac{\tau_g - \tau_f}{2} \quad (5)$$

Parameter tambahan $\tau_{tf} = \tau_t + \tau_f$ dan sistem matriks A_t yang didapat dari Persamaan 4 dan 5 diperkenalkan untuk penyederhanaan [4].

$$A_t = \begin{bmatrix} \tau_{fg} & \varepsilon & \tau_{tf} + \tau_{fg} & \varepsilon & \tau_{tf} + \tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon \\ \tau_g & \varepsilon & \tau_{tf} + \tau_g & \varepsilon & \tau_{tf} + \tau_g & \varepsilon & \tau_g & \varepsilon \\ \tau_{tf} & \varepsilon & \tau_{fg} & \varepsilon & \varepsilon & \varepsilon & \tau_{tf} & \varepsilon \\ \tau_t & \varepsilon & \tau_g & \varepsilon & \varepsilon & \varepsilon & \tau_t & \varepsilon \\ \tau_{tf} & \varepsilon & \varepsilon & \varepsilon & \tau_{fg} & \varepsilon & \tau_{tf} & \varepsilon \\ \tau_t & \varepsilon & \varepsilon & \varepsilon & \tau_g & \varepsilon & \tau_t & \varepsilon \\ \tau_{fg} & \varepsilon & \tau_{tf} + \tau_{fg} & \varepsilon & \tau_{tf} + \tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon \\ \tau_g & \varepsilon & \tau_{tf} + \tau_g & \varepsilon & \tau_{tf} + \tau_g & \varepsilon & \tau_g & \varepsilon \end{bmatrix} \quad (6)$$

Dengan mengganti parameter $\tau_g = 3s$ dan $\tau_f = 1s$, maka matriks A_t pada Persamaan 6 menjadi:

$$A_t = \begin{bmatrix} 4 & \varepsilon & 6 & \varepsilon & 6 & \varepsilon & 4 & \varepsilon \\ 3 & \varepsilon & 5 & \varepsilon & 5 & \varepsilon & 3 & \varepsilon \\ 2 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon \\ 1 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & 2 & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & 1 & \varepsilon \\ 4 & \varepsilon & 6 & \varepsilon & 6 & \varepsilon & 4 & \varepsilon \\ 3 & \varepsilon & 5 & \varepsilon & 5 & \varepsilon & 3 & \varepsilon \end{bmatrix} \quad (7)$$

Menyelesaikan sistem $x(k+1) = A_t \otimes x(k)$ dari Persamaan 7 ketika $x(0) = \mathbf{0}$ dengan menggunakan bantuan Matlab dengan semua waktu kejadian diinisialisasi ke nol dan 8 iterasi yang dapat dilihat pada Tabel 1, maka diperoleh:

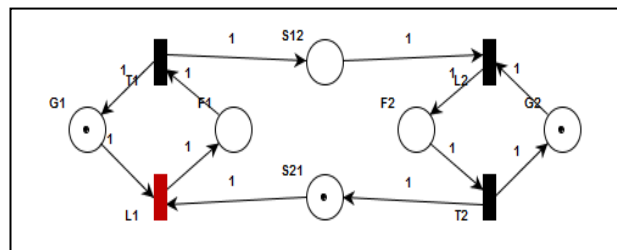
Tabel 1. Iterasi hasil pemodelan sistem menggunakan aljabar *max plus*

Iterasi	t_1	l_1	t_2	l_2	t_3	l_3	t_4	l_4
0	0	0	0	0	0	0	0	0
1	6	5	4	3	4	3	6	5
2	10	9	8	7	8	7	10	9
3	14	13	12	11	12	11	14	13
4	18	17	16	15	16	15	18	17
5	22	21	20	19	20	19	22	21
6	26	25	24	23	24	23	26	25
7	30	29	28	27	28	27	30	29
8	34	33	32	31	32	31	34	33
:	:	:	:	:	:	:	:	:

Pada iterasi pertama, terlihat bahwa pada detik ke-3 terjadi *lift off* kaki 'i' kemudian pada detik ke-4 kaki 'i' mengalami *touchdown*, lalu detik ke-5 kaki 'ii' *lift off* yang kemudian *touchdown* pada detik ke-6. Begitu seterusnya sampai iterasi ke-5 dengan masing-masing kejadian memiliki senggang waktu 1 detik sesuai dengan nilai τ_p yang diberikan.

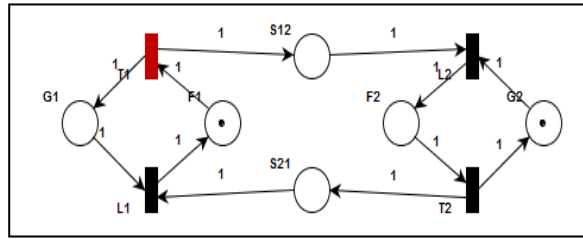
Agar lebih mudah dalam membayangkan proses gerakan pada cara berjalan *trotting*, berikut ini simulasi kejadian pada Petri net cara berjalan *trotting* dengan menggunakan *software PIPE* [3]:

- 1) Gerakan pertama pada Gambar 8 adalah ketika kaki 'i' siap melakukan *lift off*, sehingga pada Petri net transisi L_1 siap untuk di-*fire*.



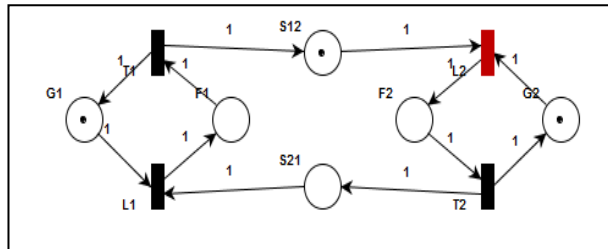
Gambar 8. kaki 'i' siap untuk melakukan *lift off*.

- 2) Setelah L_1 di-*fire*, maka yang terjadi adalah kaki 'i' berada di udara atau dalam kondisi *flight* sehingga pada Petri net terlihat bahwa pada Gambar 9 token berpindah ke *place* F_1 dan transisi T_1 siap untuk di-*fire* atau kaki 'i' siap untuk melakukan *touchdown*.



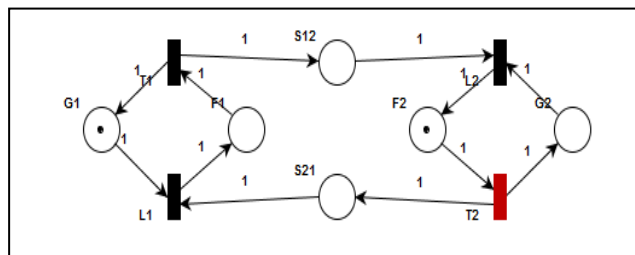
Gambar 9. Token berpindah ke *place* F_2 dan transisi T_1 siap untuk di-fire atau kaki 'i' siap untuk melakukan *touchdown*.

- 3) Setelah T_1 di-fire, selanjutnya kaki 'i' berada di tanah sehingga sekarang giliran kaki 'ii' yang siap untuk melakukan *lift off*. Terlihat bahwa pada Gambar 10 token berpindah ke *place* S_{12} untuk menunjukkan bahwa kaki 'ii' hanya boleh *lift off* setelah kaki 'i' telah menyentuh tanah yang mana sudah terpenuhi karena satu token sudah berada di G_1 .



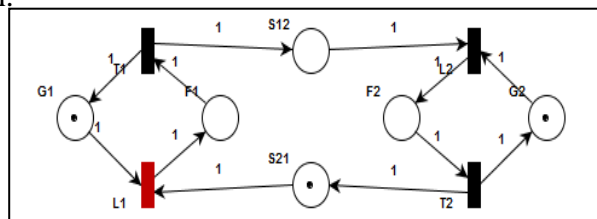
Gambar 10. Token berpindah ke *place* S_{12} .

- 4) Setelah L_2 di-fire, maka kaki 'ii' berada di udara yang terlihat dari berpindahnya token ke F_2 . Transisi berikutnya yang siap di-fire pada Gambar 11 adalah T_2 karena kaki 'ii' siap melakukan *touchdown*.



Gambar 11. Kaki 'ii' berada di udara yang terlihat dari berpindahnya token ke F_2 .

- 5) Setelah T_2 di-fire, maka kondisi robot pada Gambar 12 kembali ke gerakan pertama yakni kaki 'i' siap untuk melakukan *lift off* karena kaki 'ii' telah menyentuh tanah.



Gambar 12. Kondisi robot kembali ke gerakan pertama yakni kaki 'i' siap untuk melakukan *lift off*.

Seperti yang telah disebutkan sebelumnya bahwa kaki 'iii' memiliki waktu kejadian *lift off* dan *touchdown* yang sama dengan kaki 'i' sedangkan kaki 'iv' sama dengan kaki 'ii', maka Petri net di atas sudah sesuai dengan model gerak berjalan *trotting* robot berkaki 4 pada Gambar 2.

3 Kesimpulan

Kesimpulan dari penelitian ini adalah gerak robot empat kaki bisa dimodelkan dengan menggunakan aljabar max plus dan memudahkan pencarian kaki mana yang akan naik dan yang akan turun. Saran untuk pengembangan ke depannya lebih dikembangkan lagi terhadap model gerak robot dengan kaki lebih dari empat.

4 Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Universitas Islam Darul 'Ulum Lamongan dalam pengembangan kemampuan penelitian berkelanjutan dan wadah penelitian, serta dukungan dari mahasiswa lain dan orang-orang sekitar yang ikut banyak andil dalam proses pembuatan penelitian ini agar berjalan dengan baik dan lancar.

Daftar Pustaka

- [1] Lopes, G.D, dkk. 2009. Switching max-plus models for legged locomotion. *IEEE International Conference on Robotics and Biomimetics*. China. 221–226.
- [2] Erden, M. S., dan Leblebicioglu, K. 2008. Free gait generation with reinforcement learning for a six-legged robot. *Robotics and Autonomous Systems*. Vol. 56. 199-212.
- [3] Holmes, P., dkk. 2006. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM review*. Vol. 48. 2 : 207-304.
- [4] Subiono. 2012. *Aljabar Maxplus dan Terapannya*. Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, ITS. Surabaya.
- [6] Boom, T. V., dan de Schutter, B. 2006. Modelling and control of event discrete systems using switching max-plus-linear systems. *Control Engineering Practice*. Vol.14. 10 : 1199-1211.